

Shard key

Eligiendo una sharding key



Sharding key

- Esta clave no es más que un campo de **MongoDB** (o varios, en realidad), que nos permite decidir en qué servidor debe almacenarse el documento.
- El encargado de esta decisión es un proceso conocido como *mongos*, que recibe las peticiones y las envía al servidor correcto.

Sharding key

- Elegir una **sharding key**, es seguramente la parte más importante cuándo queremos habilitar el *sharding*.
- El rendimiento de la base de datos dependerá de la clave que elijamos.
- Eliminar o modificar una **sharding key** una vez establecida puede ser una experiencia poco recomendable.
- Es importante destacar, que los campos que se incluyan en una **sharding key** deberán tener un índice.

Puntos a tener en cuenta a la hora de elegir nuestra sharding key

- Lo primero a tener en cuenta es **qué queremos mejorar con el sharding?**

La latencia de las escrituras: quizá nos interese dividir los datos de forma geográfica. Es decir, escribir en un servidor cercano.

Si en cambio lo que queremos es mejorar la velocidad de escrituras o lecturas indistintamente, buscaremos dividir los datos de forma paralela. Así la carga se reparte equitativamente entre los servidores.

Puntos a tener en cuenta a la hora de elegir nuestra sharding key (2)

- Lo primero a tener en cuenta es **qué queremos mejorar con el sharding?**

Buscar la optimización de recursos. Por ejemplo optimizar la RAM. Para ello deberemos tener un conjunto de registros pequeño por cada servidor. Es decir, tendremos más servidores, con menos potencia, pero con menos datos que manejar. Eso sí, sin olvidarnos de cuántos *shards* (servidores) necesitamos realmente. Tener muchos incrementará la complejidad.

Puntos a tener en cuenta a la hora de elegir nuestra sharding key (3)

- Lo primero a tener en cuenta es **qué queremos mejorar con el sharding?**

Tener en cuenta el tipo de consultas que vamos a realizar. Para que el particionado sea efectivo, lo ideal es que las consultas se realicen sólo sobre uno de los *shards*. Esto implica que en las consultas deberemos incluir la **sharding key**. Si por ejemplo nuestra clave es el nombre de usuario, pero solo realizamos las búsquedas por fecha, la consulta se ejecutará en todos los servidores, perdiendo algunas de las ventajas del *sharding*.

Tipos de sharding key

- Aunque la clave puede ser cualquier campo de nuestros documentos, podríamos decir que los tipos de claves los podemos englobar en los siguientes: claves ascendentes, aleatorias, basadas en localización y compuestas.

Claves ascendentes

- Un campo que va creciendo con cada inserción.
- ObjectId o fecha.
- Siempre grabará en el último shard, lo cual provoca constantes movimientos entre shard para balancear el sistema.

Claves aleatorias

- Son campos que tienen un valor aleatorio, y son únicos en cada inserción.
- GUID, MD5
- Las escrituras se van repartiendo de forma homogénea entre los distintos servidores. De esta manera se reducen mucho los movimientos de documentos entre *shards*.
- Lo malo de estas claves, es que hacer consultas sobre ellas no siempre es fácil. Como hemos dicho antes, lo ideal es que en la consulta vaya la clave.

Claves basadas en localización

- IP, latitud, longitud
- Añadir tags: Los *tags* permiten indicar de forma manual los rangos mínimo y máximo que tiene cada shard. Así podríamos añadir un rango de IP que vaya de la 192.0.0.0 a la 192.255.255.255 de manera que los documentos cuyo campo IP pertenezcan a ese rango, queden todos en el mismo servidor.

Claves compuestas

- Varios datos en el documento.
- Como ejemplo podemos pensar en una tienda online. La **shard key** estará compuesta por nombre de usuario (aleatorio) y el identificador del pedido (incremental). De esta manera los pedidos se distribuirán de forma homogénea al ser el usuario aleatorio, pero los datos de un mismo usuario estarán concentrados en pocos *shards*, lo que mejorará el tiempo de búsqueda de los pedidos.

